IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Steven W. Githens et al.	§	Filed: September 11, 2003
Serial No.: 10/660,143	§	Group Art Unit: 2179
Confirmation No.: 4972	§	Examiner: Jordany Nunez

For: RICH GRAPHIC VISUALIZATION GENERATION FROM ABSTRACT DATA REPRESENTATION

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Jordany Nunez, or electronically transmitted via EFS-Web, on the date shown below:

May 1, 2008 /John C. Garza/
Date John C. Garza

Dear Sir:

APPEAL BRIEF-RESUBMISSION IN RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF DATED APRIL 1, 2008

Applicants re-submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2179 dated April 3, 2007, finally rejecting claims 1-4, 7-25 and 28-47. The final rejection of claims 1-4, 7-25 and 28-47 is appealed. This Appeal Brief is believed to be timely since it is transmitted by the due date of May 1, 2008, as set by the mailing of the Notice of Non-Compliant Appeal Brief Dated April 1, 2008. Applicants believe that the Appeal Brief is in compliance of the requirements under 37 CFR 41.37(c).

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-4, 7-25 and 28-47 are pending in the application. Claims 1-47 were originally presented in the application. Claims 5-6 and 26-27 have been canceled without prejudice. Claims 1-4, 7-25 and 28-47 stand finally rejected as discussed below. The final rejections of claims 1-4, 7-25 and 28-47 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner, including amendments to the claims proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments include methods (see claims 1-4 and 7-21), a computer program stored on computer readable storage media (see claims 22-25, 28-42), and computers (see claims 43-47) directed to generating a graphical representation of data. More specifically, methods, systems and articles of manufacture for generating graphics rendering language for graphical representations of data are provided. The methods generally comprise generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data, and generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language. *See Application*, page 11, lines 3-25; *Abstract*. For a description of the physical environment of the invention, see *Application*, p. 13-16, for a description of the software environment of the invention, see *Application*, p. 17-18, for a description of methods for generating graphical representations of data, see *Application*, p. 29-31, and for a description of graphic interfaces utilizing the invention, see *Application*, p. 18-29.

A. CLAIM 1 - INDEPENDENT

Claim 1 recites a computer-implemented method of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; FIG. 5. As claimed, the method includes generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The method also includes selecting a subset of the plurality

of subsets of transformation rules in accordance with a requested graphical representation type. See *Application*, page 26, line 6 – page 26, line 27; FIG. 2. The method also includes generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language; wherein generating the concrete data structure is done by operation of a computer processor. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

B. CLAIM 11 - INDEPENDENT

Claim 11 recites a computer-implemented method of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; FIG. 5. As claimed, the method includes receiving a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The method also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The method also includes generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules; wherein

generating the concrete data structure is done by operation of a computer processor. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

C. CLAIM 19 - INDEPENDENT

Claim 19 recites a method of generating an abstract data structure for a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; FIG. 5. As claimed, the method includes providing a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type. See *Application*, page 20, line 14 – page 22, line 42; FIG. 4A. The method also includes determining a requested graphical representation type. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The method also includes selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes generating an abstract data structure using the selected abstract data structure template. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language; wherein transforming the abstract data structure is done by operation of a computer processor. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

D. CLAIM 20 - INDEPENDENT

Claim 20 recites a method of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; FIG. 5. As claimed, the method includes receiving a selection of a graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The method also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract

graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language; wherein transforming the abstract data structure is done by operation of a computer processor. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

E. CLAIM 21 - INDEPENDENT

Claim 21 recites a method of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; FIG. 5. As claimed, the method includes receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The method also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The method also includes selecting transformation rules for transforming the abstract data structure into a concrete data

structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The method also includes generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

F. CLAIM 22 - INDEPENDENT

Claim 22 recites a computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; page 13, line 12 – page 14, line 8; FIG. 5. As claimed, the process includes generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The process also includes selecting a subset of the plurality of subsets of transformation rules in accordance with a requested graphical representation type. See *Application*, page 26, line 6 – page 26, line 27; FIG. 2. The process also includes generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

G. CLAIM 32 - INDEPENDENT

Claim 32 recites a computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; page 13, line 12 – page 14, line 8; FIG. 5. As claimed, the process includes receiving a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The process also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The process also includes generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

H. CLAIM 40 - INDEPENDENT

Claim 40 recites a computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating an abstract data structure for a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; page 13, line 12 – page 14, line 8; FIG. 5. As claimed, the process

includes retrieving a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type. See *Application*, page 20, line 14 – page 22, line 42; FIG. 4A. The process also includes determining a requested graphical representation type. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The process also includes selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes generating an abstract data structure using the selected abstract data structure template. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

I. CLAIM 41 - INDEPENDENT

Claim 41 recites a computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; page 13, line 12 – page 14, line 8; FIG. 5. As claimed, the process includes receiving a selection of a graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The process also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation

type. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

J. CLAIM 42 - INDEPENDENT

Claim 42 recites a computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data. See *Application*, page 11, line 3 – page 12, line 26; page 13, line 12 – page 14, line 8; FIG. 5. As claimed, the process includes receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The process also includes selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The process also includes selecting transformation rules for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The process also includes generating, on the basis of the abstract data structure, a concrete data structure

defining a concrete graphical representation in a graphics rendering language using the transformation rules. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

K. CLAIM 43 - INDEPENDENT

Claim 43 recites a computer comprising at least one processor, a memory, a database having data, and at least one graphics language framework residing in the memory for generating graphics rendering language for the data. See *Application*, page 11, line 3 – page 12, line 26; page 14, line 9 – page 18, line 7; FIG. 1-3. As claimed, the at least one graphics language framework, when executed by the at least one processor, is configured for generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The at least one graphics language framework is further configured for selecting a subset of the plurality of subsets of transformation rules in accordance with a requested graphical representation type. See *Application*, page 26, line 6 – page 26, line 27; FIG. 2. The at least one graphics language framework is further configured for generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

L. CLAIM 44 - INDEPENDENT

Claim 44 recites a computer comprising at least one processor, a memory, a database having data, and at least one abstract data structure interface residing in the memory for generating an abstract data structure for a graphical representation of the data. See *Application*, page 11, line 3 – page 12, line 26; page 14, line 9 – page 18, line 7; FIG. 1-3. As claimed, the at least one abstract data structure interface, when executed by the at least one processor, is configured for retrieving a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type. See *Application*, page 20, line 14 – page 22, line 42; FIG. 4A. The at least one abstract data structure interface is further configured for determining a requested graphical representation type. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The at least one abstract data structure interface is further configured for selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one abstract data structure interface is further configured for generating an abstract data structure using the selected abstract data structure template. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one abstract data structure interface is further configured for transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

M. CLAIM 45 - INDEPENDENT

Claim 45 recites a computer comprising at least one processor, a memory, a database having data, and at least one graphics language framework residing in the memory for generating graphics rendering language for the data. See *Application*, page 11, line 3 – page 12, line 26; page 14, line 9 – page 18, line 7; FIG. 1-3. As claimed, the at least one graphics language framework, when executed by the at least one processor, is configured for receiving a selection of a graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The at least one graphics language framework is further configured for selecting an

abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

N. CLAIM 46 - INDEPENDENT

Claim 46 recites a computer comprising at least one processor, a memory, a database having data, and at least one graphics language framework residing in the memory for generating graphics rendering language for the data. See *Application*, page 11, line 3 – page 12, line 26; page 14, line 9 – page 18, line 7; FIG. 1-3. As claimed, the at least one graphics language framework, when executed by the at least one processor, is configured for receiving a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The at least one graphics language framework is further configured for selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page

20, line 14 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in a graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The at least one graphics language framework is further configured for generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphics rendering language using the transformation rules. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

O. CLAIM 47 - INDEPENDENT

Claim 47 recites a computer comprising at least one processor, a memory, a database having data, and at least one graphics language framework residing in the memory for generating graphics rendering language for the data. See *Application*, page 11, line 3 – page 12, line 26; page 14, line 9 – page 18, line 7; FIG. 1-3. As claimed, the at least one graphics language framework, when executed by the at least one processor, is configured for receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set. See *Application*, page 18, line 8 – page 19, line 20; FIG. 4A; FIG. 7. The at least one graphics language framework is further configured for selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 20, line 14 – page 26, line 5; FIG. 2;

FIG. 4A-4B. The at least one graphics language framework is further configured for generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation. See *Application*, page 17, line 1 – page 18, line 7; page 23, line 1 – page 26, line 5; FIG. 2; FIG. 4A-4B. The at least one graphics language framework is further configured for selecting transformation rules for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type. See *Application*, page 17, line 1 – page 18, line 7; page 26, line 6 – page 28, line 2; FIG. 2; FIG. 4B. The at least one graphics language framework is further configured for generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules. See *Application*, page 28, line 3 – page 28, line 9; FIG. 2-3.

Grounds of Rejection to be Reviewed on Appeal

1. Rejection of claims 1-4, 7, 8, 11-16, 19-25, 28, 29, 32-37, and 40-47 under 35 U.S.C. § 102(e) as being anticipated by *Cox et al*. (US Patent No. 2002/0156806) (hereinafter *Cox*)

2. Rejection of claims 9, 10, 17, 18, 30, 31, 38, and 39 under 35 U.S.C. § 103(a) as being unpatentable over *Cox.*

ARGUMENTS

1. Rejection of claims 1-4, 7, 8, 11-16, 19-25, 28, 29, 32-37, and 40-47 under 35 U.S.C. § 102(e) as being anticipated by *Cox*.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Regarding claims 1, 11, 21, 22, 32, 42, 43, 46, 47:

In this case, *Cox* does not disclose "each and every element as set forth in the claim". For example, *Cox* does not disclose "*transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type*" as recited in claim 1. Claims 11, 21, 22, 32, 42, 43, 46, and 47 recite similar limitations.

In the *Final Office Action* dated 9/25/2007, the Examiner argues that the above limitations are disclosed by *Cox* page 5, paragraph 44. The Examiner analogizes "the raw data being analyzed as the abstract data structure, and the implementation of the visualization… as the concrete data structure." The Examiner also analogizes the "actions" disclosed in *Cox* to the recited "transformation rules." Such "actions" are described in *Cox* as being selectable command options, presented to a user in "a list or through a graphical user interface," which "allow a user to change view parameters, select a subset of the author's data for viewing, or select specific data for display" (*Cox*, paragraph 0044).

However, Applicants respectfully submit that the Examiner's analogy fails to explain how the cited material, as well as *Cox* generally, discloses "each and every element" of the claim. For example, the material cited by the Examiner fails to explain how a list of user command options (*i.e.*, "actions") can represent the recited *subsets of transformation rules.* Applicants respectfully submit that a user command option to "select specific data for display" is not analogous to a transformation rule. Further, even if it is assumed, *arguendo*, that a listing of user "actions" is somehow analogous to the limitation of *subsets of transformation rules*, this analogy does not conform to the remaining limitations of the present claims. For example, *Cox* does not teach that each of the listed actions (read as "subset of transformation rules") in any way *describes graphical attributes of a requested graphical representation type,* or that the "actions" are *specific to a different graphics rendering language*, as required by claim 1. Further, *Cox* does not disclose that the "actions" support *a plurality of graphical representation types and a plurality of graphics rendering languages*, as required by claim 1. Thus, the recited limitations are not disclosed by *Cox.*

Accordingly, for the foregoing reasons, Applicants submit that *Cox* does not disclose all the limitations recited by claims 1, 11, 21, 22, 32, 42, 43, 46, and 47, and therefore, respectfully request that this rejection be withdrawn.

Regarding claims 2-4, 7-10, 12-18, 23-25, 28-30, and 33-39:

Claims 2-4, 7-10, 12-18, 23-25, 28-30, and 33-39 each ultimately depend from one of claims 1, 11, 22 or 32. As Applicants believe the above remarks demonstrate that the base claims are allowable, Applicants believe that the respective dependent claims are also allowable, and allowance of these claims is respectfully requested.

Regarding claims 19, 20, 40, 41, 44, and 45:

Cox does not disclose the limitation of "*transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language*" recited in claim 19. Claims 20, 40, 41, 44, and 45 recite similar limitations. The Examiner argues that this limitation is disclosed

by *Cox*, Figure 8, paragraph [0044] and paragraph [0080]. In particular, the Examiner argues that, when a user selects a particular view, "the code language used to represent the views" represents a different graphic rendering language (See *Advisory Action* dated 12/7/2007, page 2). However, Applicants respectfully submit that the cited material in *Cox* makes no mention of any graphics rendering language, much less that each "view" is rendered in a different graphics rendering language. Thus, the recited limitation is not disclosed by *Cox.*

Further, *Cox* also does not disclose the limitation of "*abstract data structure templates, each … associated with a specific graphical representation type*" recited in claim 19. Claims 20, 32, 40, 41, 44, and 45 recite similar limitations. The Examiner argues that the recited *abstract data structure templates* are disclosed by, *e.g.*, a "BarChart view object" (*Final Office Action*, page 2). However, this analogy fails to conform to the other limitations of claim 19. That is, claim 19 requires *generating an abstract data structure using the selected abstract data structure template.* The Examiner has analogized the "raw data being analyzed as the abstract data structure" (*Final Office Action*, page 7). Thus, following the Examiner's analogy requires that the raw data ("abstract data structure") is generated using a view object such as "BarChart" ("*abstract data structure templates"*). Applicants respectfully submit that this analogy clearly does not make sense, as *Cox* does not describe "raw data" as being generated with the "view objects." Thus, the recited limitation is not disclosed by *Cox*.

Accordingly, for the foregoing reasons, Applicants submit that *Cox* does not disclose all the limitations recited by claims 19, 20, 40, 41, 44, and 45, and therefore, respectfully request that this rejection be withdrawn.

Therefore, the claims are believed to be allowable, and allowance of the claims is respectfully requested.

2. Rejection of claims 9, 10, 17, 18, 30, 31, 38, and 39 under 35 U.S.C. § 103(a) as being unpatentable over *Cox*.

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. *See* MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP § 2143. The present rejection fails to establish at least the third criterion.

Each of the claims rejected under 103(a) ultimately depend from one of base claims 1, 11, or 32. As Applicant believes the above remarks demonstrate that the base claims 1, 11, and 32 are allowable, Applicant believes that the respective dependent claims are also allowable, and allowance of these claims is respectfully requested.

Therefore, the claims are believed to be allowable, and allowance of the claims is respectfully requested.

CONCLUSION

The Examiner errs in finding that:

1. Claims 1-4, 7, 8, 11-16, 19-25, 32-37, and 40-47 are anticipated by *Cox* under 35 U.S.C. § 102(e); and
2. Claims 9, 10, 17, 18, 30, 31, 38, and 39 are unpatentable over *Cox* under 35 U.S.C. § 103(a).

Withdrawal of the rejections and allowance of all claims is respectfully requested.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellant(s)

CLAIMS APPENDIX

1. (Previously Presented): A computer-implemented method of generating a graphical representation of data, comprising:

generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data;

providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages;

selecting a subset of the plurality of subsets of transformation rules in accordance with a requested graphical representation type; and

generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language; wherein generating the concrete data structure is done by operation of a computer processor.

2. (Previously Presented): The method of claim 1, wherein generating the abstract data structure comprises:

providing a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type;

determining the requested graphical representation type; and

selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type;

the abstract data structure being generated using the selected abstract data structure template.

3. (Original): The method of claim 2, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

4. (Original): The method of claim 2, wherein the plurality of abstract data structure templates is associated with a particular data source of the data.

5. (Canceled)

6. (Canceled)

7. (Previously Presented): The method of claim 1, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

8. (Original): The method of claim 1, wherein at least one of the abstract data structure and the concrete data structure is defined in Extensible Markup Language (XML).

9. (Original): The method of claim 1, wherein the concrete data structure is defined in a vector-based graphics language.

10. (Original): The method of claim 9, wherein the vector-based graphics language is one of Vector Markup Language (VML), Scalable Vector Graphics (SVG), and Hypertext Markup Language (HTML) Image Maps.

11. (Previously Presented): A computer-implemented method of generating a graphical representation of data, comprising:

receiving a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation;

providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules; wherein generating the concrete data structure is done by operation of a computer processor.

12. (Original): The method of claim 11, further comprising:

rendering the data set, as described in the graphics rendering language, in a graphic.

13. (Original): The method of claim 11, wherein the graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

14. (Original): The method of claim 11, wherein the plurality of abstract data structure templates is associated with a particular data source of the data.

15. (Original): The method of claim 11, further comprising:

selecting a subset of the transformation rules in accordance with the graphical representation type; and

generating the concrete data structure using the subset of the transformation rules.

16. (Original): The method of claim 11, wherein at least one of the abstract data structure and the concrete data structure is defined in Extensible Markup Language (XML).

17. (Original): The method of claim 11, wherein the concrete data structure is defined in a vector-based graphics language.

18. (Original): The method of claim 17, wherein the vector-based graphics language is one of Vector Markup Language (VML), Scalable Vector Graphics (SVG), and Hypertext Markup Language (HTML) Image Maps.

19. (Previously Presented): A computer-implemented method of generating an abstract data structure for a graphical representation of data, comprising:

providing a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type;

determining a requested graphical representation type;

selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type;

generating an abstract data structure using the selected abstract data structure template; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language; wherein transforming the abstract data structure is done by operation of a computer processor.

20. (Previously Presented): A computer-implemented method of generating a graphical representation of data, comprising:

receiving a selection of a graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation type; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language; wherein transforming the abstract data structure is done by operation of a computer processor.

21. (Previously Presented): A computer-implemented method of generating a graphical representation of data, comprising:

receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation;

selecting transformation rules for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules.

22. (Previously Presented): A computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data, the process comprising:

generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data;

retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages;

selecting a subset of the plurality of subsets of transformation rules in accordance with a requested graphical representation type; and

generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language.

23. (Previously Presented): The computer-readable storage medium of claim 22, wherein the process further comprises:

retrieving a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type;

determining a requested graphical representation type;

selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type; and

generating the abstract data structure using the selected abstract data structure template.

24. (Previously Presented): The computer-readable storage medium of claim 23, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

25. (Previously Presented): The computer-readable storage medium of claim 23, wherein the plurality of abstract data structure templates is associated with a particular data source of the data.

26. (Canceled)

27. (Canceled)

28. (Previously Presented): The computer-readable storage medium of claim 22, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

29. (Previously Presented): The computer-readable storage medium of claim 22, wherein at least one of the abstract data structure and the concrete data structure is defined in Extensible Markup Language (XML).

30. (Previously Presented): The computer-readable storage medium of claim 22, wherein the concrete data structure is defined in a vector-based graphics language.

31. (Previously Presented): The computer-readable storage medium of claim 30, wherein the vector-based graphics language is one of Vector Markup Language (VML), Scalable Vector Graphics (SVG), and Hypertext Markup Language (HTML) Image Maps.

32. (Previously Presented): A computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data, the process comprising:

receiving a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the requested graphical representation type;

generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation;

retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules.

33. (Previously Presented): The computer-readable storage medium of claim 32, wherein the process further comprises:

rendering the data set, as described in the graphics rendering language, in a graphic.

34. (Previously Presented): The computer-readable storage medium of claim 32, wherein the graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof.

35. (Previously Presented): The computer-readable storage medium of claim 32, wherein the plurality of abstract data structure templates is associated with a particular data source of the data.

36. (Previously Presented): The computer-readable storage medium of claim 32, wherein the process further comprises:

selecting a subset of the transformation rules in accordance with the graphical representation type; and

generating the concrete data structure using the subset of the transformation rules.

37. (Previously Presented): The computer-readable storage medium of claim 32, wherein at least one of the abstract data structure and the concrete data structure is defined in Extensible Markup Language (XML).

38. (Previously Presented): The computer-readable storage medium of claim 32, wherein the concrete data structure is defined in a vector-based graphics language.

39. (Previously Presented): The computer-readable storage medium of claim 38, wherein the vector-based graphics language is one of Vector Markup Language (VML), Scalable Vector Graphics (SVG), and Hypertext Markup Language (HTML) Image Maps.

40. (Previously Presented): A computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating an abstract data structure for a graphical representation of data, the process comprising:

retrieving a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type;

determining a requested graphical representation type;

selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type;

generating an abstract data structure using the selected abstract data structure template; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language.

41. (Previously Presented): A computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data, the process comprising:

receiving a selection of a graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation type; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language.

42. (Previously Presented): A computer-readable storage medium containing a program which, when executed by a processor, performs a process of generating a graphical representation of data, the process comprising:

receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type,

wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation;

selecting transformation rules for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules.

43. (Previously Presented): A computer, comprising at least one processor a memory and further comprising:

a database having data; and

at least one graphics language framework residing in the memory for generating graphics rendering language for the data, the at least one graphics language framework, when executed by the at least one processor, being configured for:

generating an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data;

retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphical representation types and a plurality of graphics rendering languages;

selecting a subset of the plurality of subsets of transformation rules in accordance with a requested graphical representation type; and

generating, on the basis of the abstract data structure and the selected subset of transformation rules, a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language.

44. (Previously Presented): A computer, comprising at least one processor a memory and further comprising:

a database having data; and

at least one abstract data structure interface residing in the memory for generating an abstract data structure for a graphical representation of the data, the at least one abstract data structure interface, when executed by the at least one processor, being configured for:

retrieving a plurality of abstract data structure templates, each abstract data structure template being associated with a specific graphical representation type;

determining a requested graphical representation type;

selecting an abstract data structure template from the plurality of abstract data structure templates on the basis of the requested graphical representation type;

generating an abstract data structure using the selected abstract data structure template; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language.

45. (Previously Presented): A computer, comprising at least one processor a memory and further comprising:

a database having data; and

at least one graphics language framework residing in the memory for generating graphics rendering language for the data, the at least one graphics language framework, when executed by the at least one processor, being configured for:

receiving a selection of a graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical

representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the selected abstract data structure template, an abstract data structure defining a logical representation of the data set graphically represented according to the selected graphical representation type; and

transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language.

46. (Previously Presented): A computer, comprising at least one processor a memory and further comprising:

a database having data; and

at least one graphics language framework residing in the memory for generating graphics rendering language for the data, the at least one graphics language framework, when executed by the at least one processor, being configured for:

receiving a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the requested graphical representation type;

generating, on the basis of the requested graphical representation type and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in a graphical representation;

retrieving transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphics rendering language using the transformation rules.

47. (Previously Presented): A computer, comprising at least one processor a memory and further comprising:

a database having data; and

at least one graphics language framework residing in the memory for generating graphics rendering language for the data, the at least one graphics language framework, when executed by the at least one processor, being configured for:

receiving abstract attributes values comprising at least a selection of a requested graphical representation type for a selected data set;

selecting an abstract data structure template from a plurality of abstract data structure templates, each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type;

generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation;

selecting transformation rules for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type; and

generating, on the basis of the abstract data structure, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.